

**Q-1)** Sizden atletlerin performansını **farklı** atletizm branşları için takip etmenizi sağlayan bir uygulama geliştirmeniz istenmektedir. Bu amaçla, **kısa mesafe koşucusu** ve **mızrak atıcısı** içeren nesne yönelimli hiyerarşiyi modellemeniz gerekmektedir. İki atlet modeli de **ad, soyad, doğum tarihi** gibi ortak özelliklere sahiptir. Ek olarak, iki sınıfa ait model de atletlerin performans verilerini **yazdıran** iki adet fonksiyon içermelidir. Bu fonksiyonlar **en son N adet performans verisini yazdıran ve en iyi sonucu yazdıran** fonksiyonlardır.

Her atlet performans sonuçlarının saklandığı kayıtlardan oluşan ve dinamik olarak yaratılan bir dizi içermelidir. Performans verisi kaydında **yarışma zamanını string, sonucu double (mesela 155 metre mızrak atışı) ve yarışma şehri string** olarak tutulmalıdır. Performans verileri aşağıdaki verilen şekilde setlenebilir.

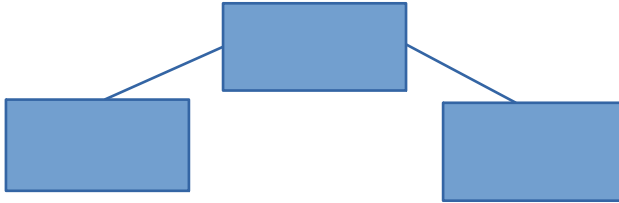
Uygulama kodu ayrıca atletlere ait performans verilerini ve en iyi zamanı yazdıran bir fonksiyonu main dosyasında gerçekleştirmelidir. Bu fonksiyon aşağıdaki gibi çağrılabilir:

```
JavelinThrower jt("Sedat", "Gormus", 15);
shortDistanceRunner sr("Mehmet", "Gormus", 15, 100);

jt.setPerformanceData("1/1/2018", 125, "Trabzon");
sr.setPerformanceData("1/1/2018", 9.95, "Trabzon");

printPerformanceData(jt);
printPerformanceData(sr);
```

**a - 10 puan)** Aşağıda verilen kutulara uygun şekilde sınıf hiyerarşisini oluşturun. Bu hiyerarşide mızrakAtıcısı ve kısaMesafeKoşucusu sınıfları olmalı ve bütün sınıfların **public, protected ve private** verileri ve metodları olmalıdır.



**b - 15 points)** Sınıfların listelerini veriniz (PS: 3 .h dosyası yazmalısınız ve sınıflar arasındaki ilişkileri düzgün şekilde belirlemelisiniz).

**c - 15 points)** Sınıf gerçekleştirme dosyalarını (.CPP) oluşturunuz (PS: her .cpp dosyası constructors (yapıcı), destructors (yıkıcı) ve gerekli fonksiyonları içermelidir.)

**d - 10 points)** (Yukarıdaki örnekte de gösterildiği gibi) her bir sınıfın yazdırma fonksiyonlarını polimorfik olarak çağırdığınız bir main fonksiyonu oluşturunuz. Fonksiyonların çıktıları aşağıdaki gibi olacaktır:

Name : Sedat Surname : Gormus

1-) Date : 1/1/2018--Trabzon--125 meters for javelin throwing

Best Performance of Sedat Gormus is achieved on: 1/1/2018--Trabzon--125 meters

Name : Mehmet Surname : Gormus

1-) Date : 1/1/2018--Trabzon--9.95 seconds for 100 meters running

Best Performance of Mehmet Gormus is achieved on: 1/1/2018--Trabzon--9.95 seconds

arka sayfayı çevirin >>>>

**e - 15 points)** mızrakAtıcı ve kısaMesafeKoşucusu sınıfları dinamik olarak oluşturulan ve performans verisi içeren dizilere sahiptir. Bu dizilerin derin kopyalamasının nasıl yapılacağını iki cümleyle açıklayınız ve derin kopyalama için gerekli operatörü sadece bir sınıf için tekrar tanımlayınız (operator overloading).

**f- 15 points)** Lütfen indexleme operatörünü ([]) mızrakAtıcısı sınıfı için tekrar tanımlayınız. Indexleme operatörü istenen indexteki performans verisini aşağıdaki gibi geri getirmelidir. Indexin dizinin sınırlarını aşmamasına dikkat edilmelidir.

```
struct performance p = jt[1];
```

Q-2) Lütfen aşağıdaki NYP kavramlarını **iki cümleyle** açıklayınız.

**a - 10 points)** Sarmalama (Encapsulation)

**b- 10 points)** Arkadaş Fonksiyonlar(Friend functions)

Bol Şans